We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Chapter

A Hybrid Approach for Solving Constrained Multi-Objective Mixed-Discrete Nonlinear Programming Engineering Problems

Satadru Roy, William A. Crossley and Samarth Jain

Abstract

Several complex engineering design problems have multiple, conflicting objectives and constraints that are nonlinear, along with mixed discrete and continuous design variables; these problems are inherently difficult to solve. This chapter presents a novel hybrid approach to find solutions to a constrained multi-objective mixed-discrete nonlinear programming problem that combines a two-branch genetic algorithm as a global search tool with a gradient-based approach for the local search. Hybridizing two algorithms can provide a search approach that outperforms the individual algorithms; however, hybridizing the two algorithms, in the traditional way, often does not offer advantages other than the computational efficiency of the gradient-based algorithms and global exploring capability of the evolutionary-based algorithms. The approach here presents a hybridization approach combining genetic algorithm and a gradient-based approach with improved information sharing between the two algorithms. The hybrid approach is implemented to solve three engineering design problems of different complexities to demonstrate the effectiveness of the approach in solving constrained multi-objective mixed-discrete nonlinear programming problems.

Keywords: multi-objective, mixed-discrete, constrained, nonlinear, genetic algorithm, gradient-based optimization

1. Introduction

Many engineering design problems require simultaneous optimization of multiple, often competing, objectives. Unlike in single-objective optimization, a multi-objective problem with competing objectives has no single solution. An optimum solution with respect to only one objective may not be acceptable when measured with respect to the other objectives. Multi-objective problems have a number of solutions called the Pareto-optimal set, named after Vilfred Pareto [1], that represent the range of best possible compromises amongst the objectives. Traditional gradient-based optimization algorithms are capable of addressing the multi-objective problems by converting the problem into a single-objective formulation.

IntechOpen

On the other hand, evolutionary algorithms (EAs)[1] are well suited for the multi-objective problems as they can evolve to a set of designs that represent the Pareto frontier in a single run of the algorithm [2, 3]. As a result, EAs often find application to address multi-objective problems. Despite the popularity of these algorithms to solve a wide range of problems, they, like all non-gradient meta-heuristic searches, have issues with computational cost and rate of convergence to the Pareto frontier. After some number of generations, the candidate solutions may begin to exhibit little or no improvement. Modified versions of these algorithms exist which improve the convergence rate [4]. However, hybridizing EAs with an efficient gradient-based algorithm may significantly improve the convergence rate and has demonstrated the ability to solve multi-objective problems more efficiently than the EA alone [3]. Hybridization of an EA with a gradient-based local search algorithm has started to gain popularity owing to its promising capabilities to address the demerits of many optimization algorithms when used independently.

The genetic algorithm (GA) [5] is a class of EA and is a well-known population-based global search algorithm. Apart from its ability to explore the design space, GA is also capable of handling both discrete and continuous type design variables. This makes the GA an ideal choice to address problems that combine both discrete and continuous variables. However, the GA, like other EAs, does not provide any proof of convergence, and the GA cannot directly enforce constraints. Commonly, constraint handling for a GA search uses a penalty approach such that the fitness function reflects the objective function value and accounts for violated constraints. This generally requires the use of penalty multipliers to adjust the "strength" with which the penalty impacts the fitness function and selecting suitable penalty multipliers is often difficult. Further, for multi-objective problems, the different scaling or magnitude of the objectives can complicate selecting appropriate penalty multipliers.

On the other hand, Sequential Quadratic Programming (SQP) [6], is a well-known gradient-based search algorithm that directly handles constraints and provides proof of convergence to local optima using Karush-Kuhn-Tucker (KKT) optimality criteria [7]. Because SQP uses gradient information, it is a computationally efficient search algorithm. However, SQP cannot handle discrete design variables or discontinuous functions and has difficulty with multi-modal functions. Therefore, both of these (GA and SQP) well-known optimization algorithms have their own pros and cons that limit their individual applicability to fully address constrained multi-objective problems that combine both continuous and discrete type design variables. Combining the GA with SQP creates a hybrid approach that improves the overall optimization process for constrained mixed-discrete nonlinear programming problems (MDNLP).

The chapter presents a combination of the two-branch tournament GA for multi-objective problems with an SQP-based local search implementation of the goal attainment problem formulation allowing an improved information sharing between the two algorithms. To the best of the authors' knowledge, there exists no work that emphasizes the process of hybridization combining an N-branch tournament selection GA with the goal attainment formulation as the local search in a compatible manner and then demonstrates application of the approach to solve a hard-to-solve constrained multi-objective, mixed-discrete nonlinear optimization problem. Later in the chapter, the hybrid approach is applied to solve a three-bar truss problem, a ten-bar truss problem, and a greener aircraft design optimization

[1] Here, the term "evolutionary algorithm" encompasses all population-based search algorithms that use features inspired by biological evolution.

problem – all representatives of constrained multi-objective, mixed-discrete nonlinear programming problem. The truss problems have basis in test problems for structural optimization, and the motivation to select a greener aircraft design optimization problem arises from the increased concern about the environmental impact of the growing air transportation system.

2. Literature review

The ability of the EAs to evolve to a Pareto-frontier as the generation progresses makes them an ideal choice for several multi-objective optimization problems. Vector Evaluated GA (VEGA), proposed by Schaffer [8] back in 1985, is one of the earlier versions of multi-objective GA. Several multi-objective EAs are developed since then including Multi-Objective Genetic Algorithm (MOGA) [9], Strength Pareto Evolutionary Algorithm [10], Non-dominated Sorted Genetic Algorithm (NSGA) [11] to mention a few popular ones.

Coello [2, 12] has conducted comprehensive literature surveys of various evolutionary multi-objective techniques. Konak et al. [13] compared various multi-objective optimization algorithms and provides a set of guidelines to follow while developing a multi-objective algorithm. Their effort primarily lies in guiding researchers with very little background in MOGA and making them familiar with the ideas and approaches of multi-objective optimization.

One such multi-objective algorithm named Non-dominated Sorting Genetic Algorithm (NSGA), developed by Srinivas and Dev [11] – arguably one of the most widely used multi-objective EAs – uses the concept of non-dominated sets originally proposed by Goldberg in his book on Genetic Algorithm and Machine Learning [14]. The NSGA approach maintains sets of non-dominated individuals, with the first set of individuals not dominated by any other individuals in the population. The second set finds the new set of non-dominated individuals after excluding the individuals from the first set. This step continues until all the individuals in the population are categorized inside the non-dominated sets.

A majority of these multi-objective algorithms, in some form, require an assignment of a scalar measure of a fitness value to the individuals in the population. As an example, MOGA [9] and NSGA [11] assign a fitness value based on a ranking scheme depending on the individual's levels of domination. The two-branch tournament selection genetic algorithm presented by Crossley et al. [15] uses a tournament selection scheme that chooses parents considering both the objectives directly in the fitness functions. The individuals are evaluated based on their fitness across both the objectives. The overall process remains the same as that of a traditional GA. However, the only difference appears in the tournament selection operator. During the tournament selection step, the algorithm selects 50% of the parents based on the fitness value associated with the first objective, that is, the individuals are evaluated solely with respect to the first objective without consideration of the other objective. These selected parents are by nature strong in objective 1, or Φ_1-strong. Similarly, the tournament selects the remaining 50% of the parents based on the fitness value associated with the second objective. This second 50% are Φ_2-strong parents. With this parent selection approach, randomly choosing the selected parents to pair off for crossover, ideally would result in the following distribution of matches: 25% $\Phi_1 - \Phi_1$ type parents, 25% $\Phi_2 - \Phi_2$ type parents, and 50% are mixed i.e., $\Phi_1 - \Phi_2$ type parents.

The hybrid approach, presented in this chapter, uses this two-branch tournament selection GA as the global search optimizer and combines with a gradient-based approach to refine the search using a novel information sharing concept in

the process of hybridization. The unique tournament selection strategy of the two-branch tournament GA allows to understand the underlying trait of the parents, i.e., if they are Φ_1 or Φ_2 strong, and this information is later leveraged during the crossover step to obtain children with certain desired traits.

Another challenge with multi-objective EAs is their ability to enforce constraints. Unlike gradient-based methods, which use constraint gradient information to guide the search in the feasible direction, no such constraint gradient is available for EAs. There have been several efforts to handle the constraints for EAs; however, not all of these methods strictly or directly enforce the problem constraints. The penalty function approach is arguably the most widely known of the various approaches to handle constraints in EAs. Assuming a minimization problem, this approach adds a penalty to the objective function when constraints are violated [14].

Another simple approach includes ignoring any infeasible design solution; because this does not differentiate between constraints that are close to the constraint boundaries and those that are far apart, this constraint handling method is inefficient.

Binh and Korn [16] suggested a method to assign fitness to individuals based on combining both the objective function vector as well as the degree to which the individual violates the constraint. Infeasible individuals are categorized into different classes based on how close or how far they are to the constraints boundaries.

Fonseca and Fleming [17] proposed a priority-based constraint handling strategy where search is first driven for feasibility followed by optimality by assigning high priority to constraints and low priority to objective functions. Although there are various techniques to "handle" constraints in EAs, "enforcing" them in a robust way is still an open issue. This is another motivation to pursue the hybrid approach that leverages the efficacy of gradient-based search to enforce the problem constraints.

Further, these population-based searches have issues with computational cost and rate of convergence to the Pareto frontier. After some number of generations, the candidate solutions may begin to exhibit little or no improvement. Modified versions of the algorithms work to improve the convergence rate [4, 18]; however, hybridizing EAs or GAs with an efficient gradient-based algorithm can significantly improve the convergence rate, thereby reducing the computational cost. Hybridization of an EA or GA with a gradient-based local search algorithm is not new. There are numerous references demonstrating how hybridization may improve the quality of the search for both single objective and multi-objective problem formulations; these include, but are not limited to, those appearing in [3, 19–32]. The local search can be considered as the local learning that takes place in an individual throughout its lifespan. Some of the approaches apply the local search to the final non-dominated set, while some techniques apply local search to all or many individuals of the population as the generation progresses.

The effort here extends the previous effort by Lehner and Crossley [27] to include a multi-objective formulation and combine the advantage of the hybrid approach with an novel information sharing technique between the global and the local search. The two-branch tournament selection GA algorithm globally explores the design space handling both discrete and continuous type variables, while the gradient-based approach sees only the continuous variables in a goal attainment formulation and seeks to efficiently refine the population based on the information passed on by the top-level GA while enforcing all the problem constraints.

3. Methodology and approach

The hybrid approach presented in this chapter combines the two-branch tournament GA (see **Figure 1**) for the global search [15] and the goal attainment SQP

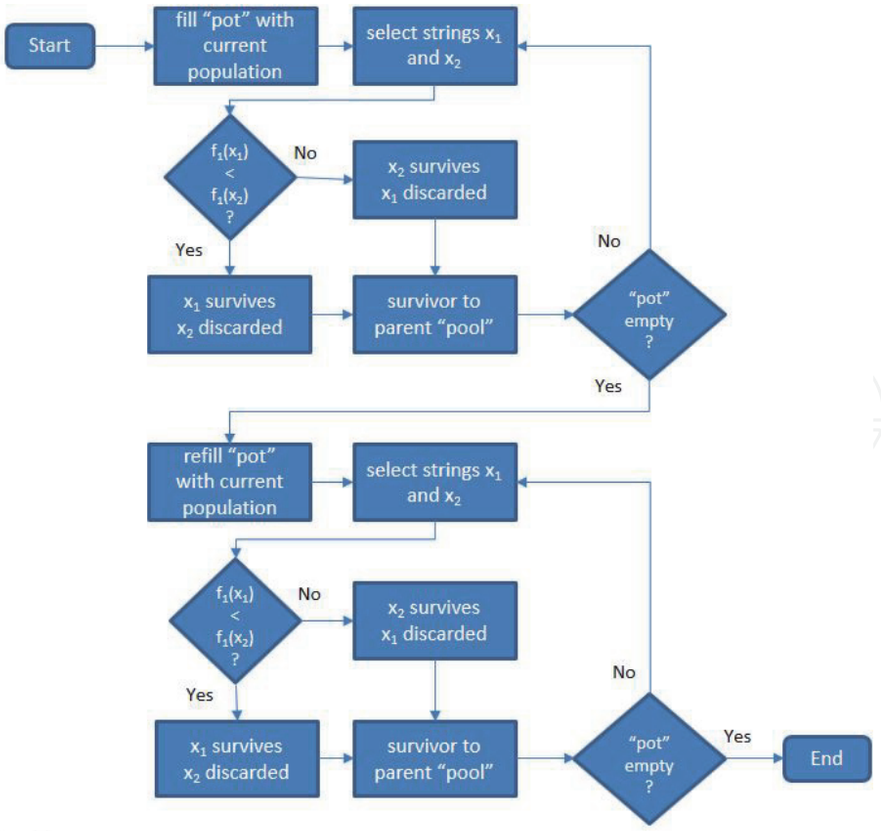


Figure 1.
Original two branch tournament selection GA for two objective problems. Adapted from reference [15].

algorithm provided in the function *fgoalattain* available from the MATLAB Optimization Toolbox [33] as the local search. For solution via hybrid approach, the problem statement contains two levels, as appears below.

3.1 Level I: Two-branch tournament genetic algorithm

The top level of the problem, which the GA sees as its optimization problem, is a bound constrained (i.e., only side constraints on the continuous design variables) multi-objective minimization problem that uses the two-branch tournament selection technique with some modification to include the local search. This level includes both discrete and continuous design variables of the original problem. The continuous variables in this level, x_c^0, are the initial values (starting point) for the local search problem. This way, the GA acts like a guide for a sequential multi-start approach as it searches the combined discrete and continuous design space. The top level formulation appears below:

$$\underset{x_d, x_c^0}{\text{Maximize}} : \left\{ \begin{array}{l} f_1(x_d, x_c^0) \\ f_2(x_d, x_c^0) \end{array} \right\}$$

$$\text{Subject to :} \tag{1}$$

$$(x_c)_i^L \leq (x_c)_i \leq (x_c)_i^U$$

$$(x_d)_i \in \text{A, B, C, D, } \ldots \text{(discrete variables)}$$

In the original two-branch tournament selection GA, the tournament step selects 50% of the parents based on the fitness value associated with the first objective.

These parents are by nature strong in objective 1 or Φ_1-strong. Similarly, the tournament selects the remaining 50% of the parents based on the fitness value associated with the second objective. This second 50% are Φ_2-strong parents. With this parent selection approach, randomly choosing the selected parents to pair off for crossover would result, on average, in the following distribution of matches: 25% Φ_1-Φ_1 type parents, 25% Φ_2-Φ_2 type parents, and 50% are mixed i.e., Φ_1-Φ_2 type parents. This has the effect of generating many compromise solutions near the middle of the Pareto frontier, potentially limiting the spread and quality of the Pareto-front. The approach described in this chapter improves the spread and quality of the Pareto front by pairing off the parents in a more prescribed manner. A flowchart depicting how the modified two-branch tournament GA interacts with the gradient-based (SQP) for local search appears in **Figure 2**.

With a given goal f_i^G, a starting point for the continuous variables x_c^0 and a set of discrete values x_d, the goal attainment problem formulation, for each individual in the GA-level population, seeks to find the optimal design x_c^*. The goal attainment problem formulation also assigns the fitness value to the individuals, thereby waiving off the need of fitness evaluation at the GA-level (level I). Using the fitness information of these populations, a new set of goals are generated for the next iteration following the tournament selection, crossover and mutation steps of the two-branch tournament GA algorithm. The resulting design x_c^* need not conform to the binary Gray coding scheme implemented to represent the chromosome of each individual in the population. The effort here employs a Lamarckian strategy [20], that updates the chromosomes of the individual to conform to the gray-coding scheme of the GA.

Figure 3 demonstrates the parent selection process of the new two-branch tournament selection GA and the goal assignment technique with a simple example. The approach starts with a population size of $8n$, where n is any positive integer (**Figure 3** assumes a population size of 8; i.e., $n = 1$). After the two-branch tournament selection process, $4n$ parents are Φ_1-strong and the other $4n$ parents are Φ_2-strong. These parent groups remain in two separate parent pools. An additional step after the two branch selection process further categorizes these parents into

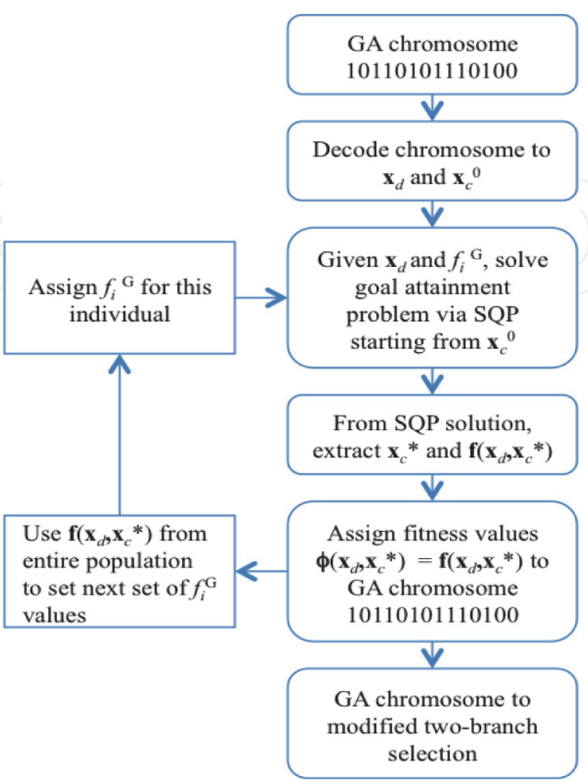


Figure 2.
Modified two branch tournament selection GA and SQP interaction.

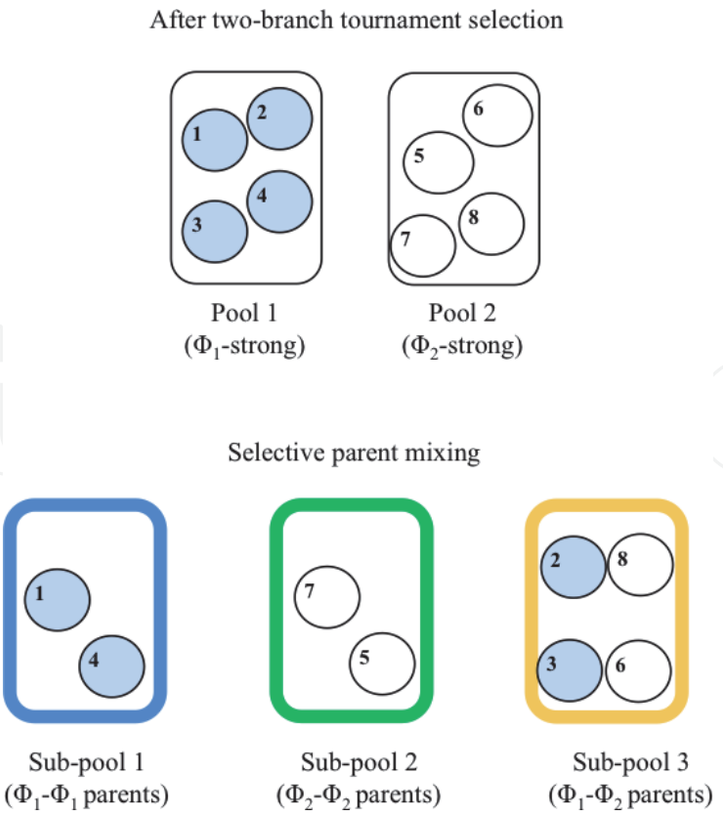


Figure 3.
Selective parent mixing strategy.

sub-pools to ensure a prescribed mix of Φ_1-strong and Φ_2- strong parents for crossover. To begin, half of the parents from pool 1 (which contains Φ_1-strong parents) are randomly moved to sub-pool 1. This sub-pool contains Φ_1-Φ_1 type parents paired for crossover and leads to offspring that will likely be Φ_1-strong. Similarly, half of the parents from pool 2 are randomly moved to sub-pool 2 to form Φ_2-Φ_2 type paired parents. Sub-pool 3 pairs parents so that a Φ_1-strong parent and a Φ_2-strong parent form children via the crossover operation. These would create children that have features from both Φ_1 and Φ_2 strong parents. This modification to the original two-branch tournament selection approach leads to a more prescribed, yet diversified, set of parents in each pool for the crossover, somewhat analogous to the idea of breeding for plant hybridization.

3.2 Level II: sequential quadratic programming

The lower-level problem presented to the SQP algorithm refines the population of the GA by searching the continuous variable space and helps the hybrid algorithm converge to the Pareto frontier at a faster rate. The *fgoalattain* algorithm, available in MATLAB, converts the multi-objective algorithm into a single-objective optimization problem by converting all the objectives into a set of inequality constraints and minimizes a slack variable γ (also called the attainment factor) as the objective.

$$\begin{aligned} \text{Given} &: x_c^0, x_d, f_i^G \\ \text{Minimize} &: \gamma \\ \text{Subject to} &: f_i(x_c) - \alpha_i \gamma \le f_i^G \\ & g_j(x_c) \le 0 \\ & h_k(x_c) = 0 \\ & (x_c)_i^L \le (x_c)_i \le (x_c)_i^U \end{aligned} \tag{2}$$

This goal attainment formulation seeks to attain values for the objectives close to a set of predefined goal values, f_i^G, without violating any of the problem constraints $g_i(x) \leq 0$ and $h_k(x) = 0$. The weight values, α_i, are set as the absolute of the corresponding goal values, f_i^G, based on the guidance in [34]. This prevents scaling issues with objectives of various dimensions and magnitudes. The solution to this problem describes the set of continuous variables x_c^* that minimizes γ and satisfies all constraints; the values of $f_i(x_c^*)$–the fitness value of the individual in the population–are returned to the GA-level for the use in the two-branch tournament selection.

The *fgoalattain* formulation needs a defined goal point in the objective space, and the algorithm tries to find a design as close as possible to these goal values. **Figure 4** illustrates the goal point assignment task for each newly created individual in the sub-pools following the example presented in **Figure 3**. In **Figure 4**, the points indicate the child "designs" from a set of parents; e.g., $C1_{(1-4)}$ is the first child from the crossover of parent 1 and parent 4. The color of the symbol indicates the parent sub-pool from which the child designs were generated. Therefore, in this example, there are two children generated from parents 1 and 4 in sub-pool 1, which are indicated with the light blue color to match **Figure 3**. There are four children generated from sub-pool 2 and two children from sub-pool 3.

To assign the goal point values, the hybrid approach first identifies the local ideal point in each generation. This ideal point is the combination of the lowest f_1 and f_2 values in the current population. For this effort, the utopia point (which includes some tolerance to give the utopia smaller–or better–f_1 and f_2 values than the ideal point with the intent of encouraging under achievement in the goal attainment problem) is set as 0.95 times the local ideal point. In subsequent generations, any new objective value smaller than the corresponding value in the current utopia point replaces that current value in the utopia point. This makes the utopia point dynamic with each generation. For two-objective problems, two perpendicular lines originate from the utopia point and extend infinitely into the objective space. These straight lines appear as dashed lines in **Figure 4**.

To assign a goal point to an individual, the approach defines a vector that originates from an individual and ends to where the vector intersects with either of the dotted lines. The point of intersection becomes the goal point for that individual. Children of parents from sub-pool 1, the Φ_1-strong sub-pool, receive a goal vector with slope of zero in the objective space. These are the horizontal arrows in **Figure 4**.

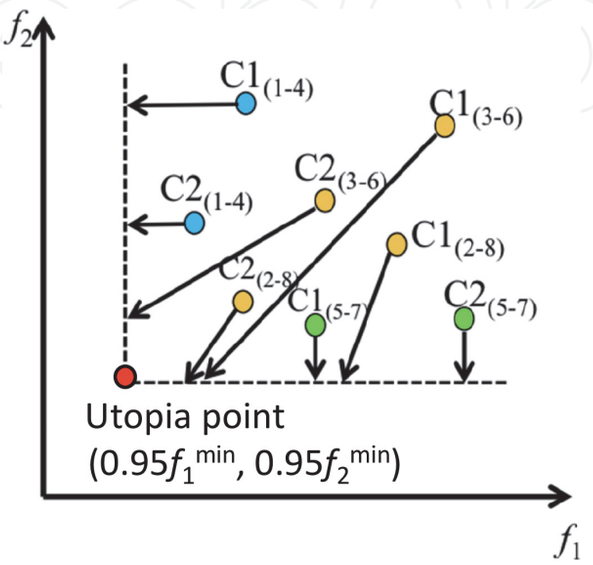


Figure 4.
Goal assignment technique.

These goal points would seek the most improvement along the direction of objective 1. Similarly, children of parents from sub-pool 2 receive a goal vector with 90 degree slope in the objective space. This ensures improvement along the direction of objective 2. Lastly, the children from sub-pool 3 parents receive goal vectors relative to their spatial location in the objective space. An individual closer towards objective 1 will have a vector inclined more towards improvement in objective 1 and vice versa.

Referring back to **Figure 3**, parents 1 and 4 from sub-pool 1 create children $C1_{(1-4)}$ and $C2_{(1-4)}$. This indicates the first child of parent 1 and 4 and the second child of parent 1 and 4 respectively. During the SQP search, these children have goal points that will minimize along the direction of f_1 without increasing their current values of f_2. $C1_{(5-7)}$ and $C2_{(5-7)}$ result from sub-pool 2, and the local search will seek to improve f_2 without increasing f_1. $C1_{(3-6)}$, $C2_{(3-6)}$, $C1_{(2-8)}$ and $C2_{(2-8)}$ all result from sub-pool 3, and they will have different goal points for their local searches to improve both f_1 and f_2. This modified parent selection and goal assignment strategy, via the hybrid formulation, seeks to exploit the tournament selection process of the two-branch tournament GA and tailor the local search for children, depending on traits of their parents.

Although the approach seems robust in enforcing constraints via goal attainment formulation, there may be instances when no feasible solution exists to the goal attainment formulation for a given set of discrete variables. In such cases, the local search will not be able to return a feasible solution and the fitness function receives a severe penalty in the GA-level in an effort to discard such discrete design choices from the population. This severe penalty has some resemblance to the approach of ignoring infeasible designs that was criticised above; however, because the situation where no locally-feasible design exists results from a specific combination of discrete variables, there is no analog to having a "nearly feasible" design with a slightly violated constraint. Severely penalizing such infeasible designs for certain combinations of discrete variable choices, in this context, is appropriate.

4. Application to engineering design problems

To demonstrate the efficacy of the hybrid approach in solving constrained multi-objective MDNLP problems, we solve three different engineering test problems with varying difficulties - a three-bar truss, a ten-bar truss, and greener aircraft design problem.

4.1 Three-bar truss problem

For the three-bar truss problem (see **Figure 5**), the problem formulation includes the objectives of minimizing the weight of the truss and minimizing the deflection of the free node. The deflection of a node is calculated as the resultant of the deflections in both the x and y directions. The problem consists of six design variables, of which three are continuous and three are discrete. The continuous variables describe the cross-sectional area of the three bars while the discrete variables describe the material selection properties of these bars. The details of the continuous design variables and their design bounds appear in **Table 1**. For this problem, four discrete material selection choices are available for each element and include aluminum, titanium, steel, and nickel options. The yield stress for every bar acts as a constraint for the problem (total three constraints), not allowing the stress in the bar to go beyond that upper limit. References [35, 36] provide more details about the three-bar truss problem. For the hybrid approach, the GA population is limited to 8 individuals while setting the upper limit for the number of generations

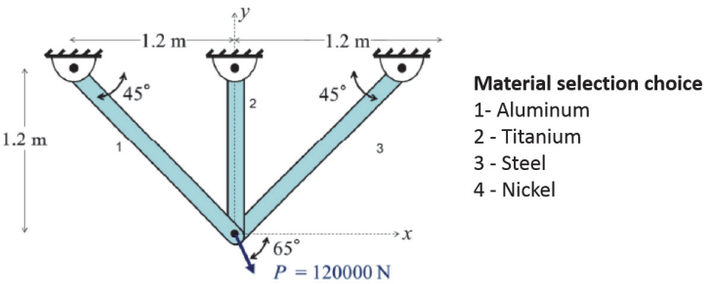


Figure 5.
Three bar truss problem.

Design variables	Lower bound	Upper bound
Cross-sectional area of bar 1 [cm^2]	0	5
Cross-sectional area of bar 2 [cm^2]	0	5
Cross-sectional area of bar 3 [cm^2]	0	5

Table 1.
Continuous variables for three-bar truss problem.

to 50. The probability of crossover is set to 0.5 and the mutation rate is fixed at 0.005. The continuous and discrete variables uses 8 and 2 bits respectively in the Gray-coded binary scheme.

The resulting Pareto frontier for the three-bar truss problem appears in **Figure 6(a)**. The plot shows the Pareto frontier has a good spread, leading to a total of 248 non-dominated points as solutions to the optimization problem. The visible trend in the non-dominated design set indicates that as the weight of the three bar truss system increases, they are accompanied by similar increases in the cross-sectional area of the bars with the material selection choice gradually shifting to steel for all the three bars. Aluminum or nickel never appeared as the material selection choice in the first two bars. The designs visible in the top left corner of the Pareto front in **Figure 6(a)** correspond to high displacement and low weight designs. The separated cluster of points (six designs) visible at the bottom right corner of the Pareto frontier corresponds to low displacement and high weight designs, with the maximum weight design having a material combination of all steel bars.

For the three-bar truss problem, only 64 possible combinations of discrete design variables exist. Hence, it is possible to perform a complete enumeration of the discrete design space and get a sense of the shape of the true Pareto front and help assess the performance of the hybrid approach. This led the authors to compare the hybrid approach (and the original two-branch tournament selection GA[2]) with a gradient-based weighted sum approach for this three-bar truss problem. The weighted sum approach converts the multi-objective problem formulation into a single objective problem by assigning weights to both the objectives and solves the single objective problem with the gradient-based approach using MATLAB's *fmincon* solver [33].

First, the objectives are normalized using the utopia point. Next, objective 1 is assigned a weight w that varies from 0 to 1 in a step increment of 0.05. The weight for the second objective is set to $1 - w$. For each possible combination of discrete

[2] The original two-branch tournament selection GA was proposed for unconstrained problems. In this example, the problem constraints in the original two-branch tournament selection GA (used for comparison) are handled using an exterior penalty approach.

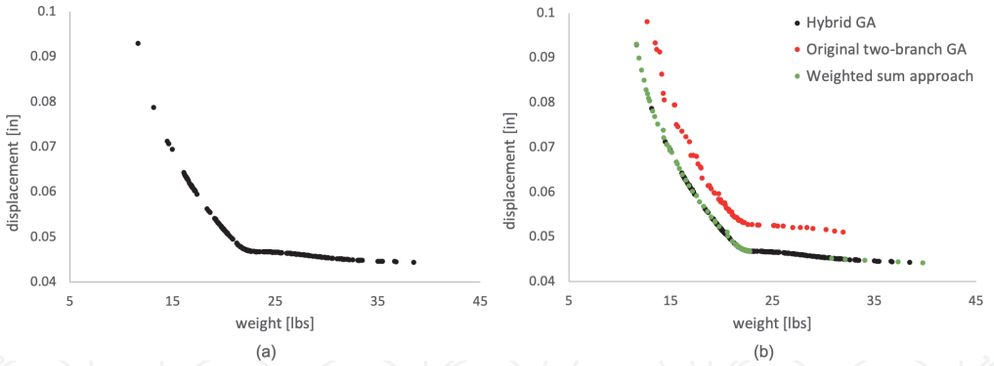


Figure 6.
Pareto front for the three-bar truss problem and its comparison with the other approaches. (a) Pareto front for the three-bar truss problem using the hybrid approach. (b) Comparison of Pareto frontier obtained using the hybrid approach, a weighted sum approach and the original two-branch tournament GA approach.

variable choice and a given weight pair, the approach leads to a single point in the objective space. The weighted sum approach then conducts gradient-based search for all 21 different weight pairs corresponding to each of the 64 possible discrete combination choices. The resulting Pareto frontier using the weighted sum approach is compared with the hybrid approach and the original two-branch tournament GA approach in **Figure 6(b)**. The original two-branch tournament GA finds an inferior set of solutions, possibly due to the lack of local search feature, and the set of solutions also has a reduced spread across the Pareto frontier. On the other hand, the weighted sum approach with complete enumeration on the material selection choices has a slightly better spread compared to the hybrid approach but with fewer non-dominated points.

Figure 7 compares how the Pareto frontier evolved with generations using the original two-branch tournament GA and the proposed hybrid approach. As expected, without the local search feature, the original two-branch tournament selection GA shows distinct improvement in both the quality and the spread of the Pareto front as the generation progresses. That is, the black diamonds (non-dominated set after second generation) are replaced with better non-dominated designs as the generation progresses. However, in the hybrid case, we start to see the shape of the final Pareto front immediately after the second generation. As the generation progresses further, more points get added to the list of non-dominated designs. This is due to the multi-start approach where the top-level GA populates various possible combinations of the discrete material selection choices and the local gradient-based search then improves these designs by varying the continuous design variables. The hybrid approach is able to rapidly get to the final Pareto front

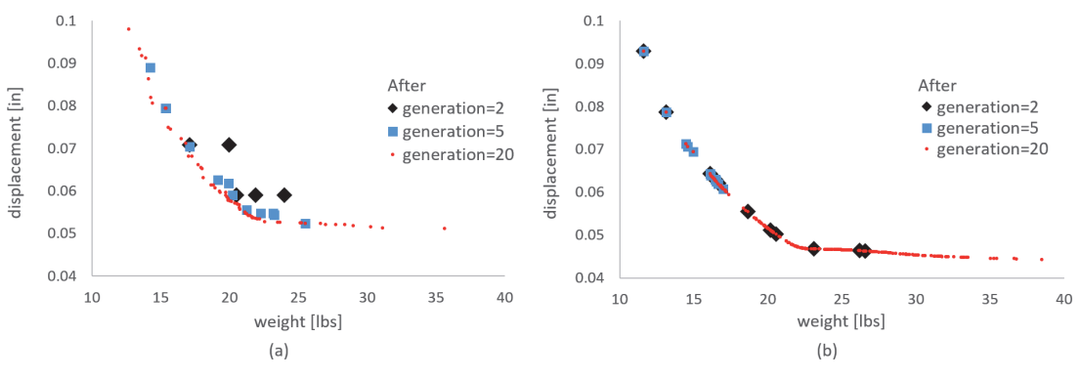


Figure 7.
Evolution of the non-dominated sets as the generation progresses. (a) Original two-branch tournament selection GA. (b) Proposed hybrid approach.

at the expense of increased number of function evaluations needed by the gradient-based local search.

4.2 Ten-bar truss problem

Next, the hybrid approach solves a more difficult and challenging version of the three-bar truss problem – a ten-bar truss. Similar to the three-bar truss problem, the ten-bar truss has the competing objectives that include minimizing the weight of the ten-bar truss system and minimizing the resultant displacement of any of the free nodes. The displacement is taken as the absolute of the maximum calculated displacement among all the bar elements. This problem consists of twenty design variables – ten continuous type and ten discrete type. The continuous variables describe the cross-sectional diameters of the ten bars, ranging from 0.1 cm^2 to 40 cm^2, while the discrete variables specify the material selection properties of these bars. Like the three-bar problem, the four discrete material choices available for each bar include aluminum, titanium, steel, and nickel. However, this problem has over one million possible combinations of the discrete choices ($4^{10} = 1,048,576$) making complete enumeration of the discrete design space computationally prohibitive, unlike the three-bar truss. References [35, 36] provide more details about the ten-bar truss problem considered in this study.

Figure 8(a) compares the Pareto front obtained using the hybrid approach after 20 GA generations with the Pareto frontier obtained using the two-branch tournament selection GA after 100 generations. The figure shows both the approaches performed well for this problem with the two-branch tournament selection GA resulting a better spread in the low weight/high displacement region of the objective space, whereas the hybrid GA has a better spread in the low displacement/high weight region. **Figure 8(b)** shows how the non-dominated set evolved as the generation progresses using the hybrid approach. We see a similar trend as that of the three-bar truss problem. That is, there is not much significant change in the final shape of the Pareto front other than the increase in the number of non-dominated designs as the generation progresses. However, this time there is slight improvement in the quality of the Pareto front (the red non-dominated set obtained after generation 20 is slightly better than the blue or the black non-dominated designs obtained at generation 5 and 2 respectively).

For the three-bar example, a majority of the improvements across the objective space are due to the gradient-based local search's ability to obtain designs with better cross-sectional area. With only 64 possible material selection combinations,

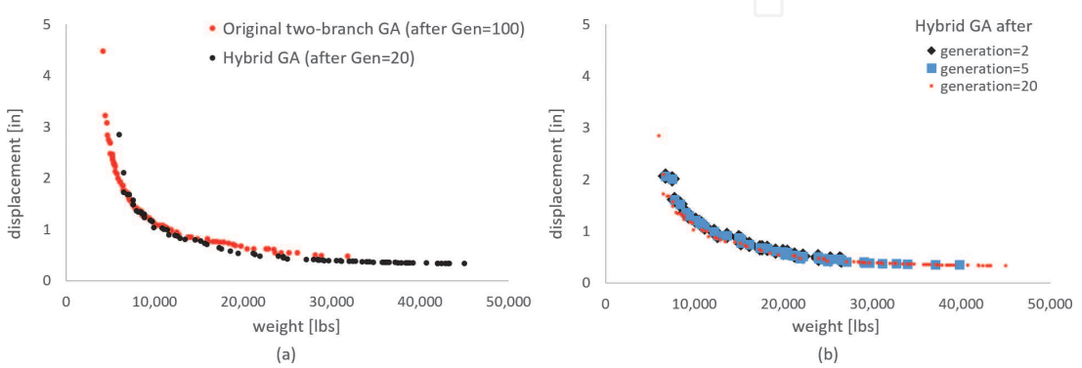


Figure 8.
Ten bar truss problem results. (a) Comparison between the original two-branch tournament GA (after 100 GA generation) and the hybrid approach (after 20 GA generations) for the 10 bar truss problem. (b) Evolution of non-dominated set as the generation progresses for the 10 bar truss problem using the hybrid approach.

there are not many discrete material selection options to explore. On the other hand, for the ten-bar truss problem, a vast majority of the improvement is due to the ability of the GA to find a better material selection combination rather than fine-tuning the cross sectional variables. It is not possible to seek further improvement in the Pareto front just by varying the continuous variables, so the local search saturates as appear in the case of black (diamonds) and blue (squares) non-dominated designs. After few more GA iterations, the algorithm is able to find better combinations of material selection that lead to further improvement in the Pareto front (red dots).

4.3 Greener aircraft design problem

The third application problem solved using the hybrid approach is the greener aircraft design problem. Here, a "greener" aircraft design problem provides an example to demonstrate the efficacy of the hybrid algorithm and its ability to solve such MDNLP problems. The intent is to find aircraft designs that represent the best possible trade-offs among performance, economics, and environmental metrics which essentially makes this a multi-objective problem. Further, with the inclusion of discrete technologies, the problem becomes MDNLP in nature.

The aircraft design optimization problem employs the NASA sizing code FLOPS [37] to evaluate discrete design configurations and perform the sizing and performance calculations of the candidate aircraft designs. The sizing code accepts both continuous and discrete design variables as input and returns the aircraft gross weight along with environmental metrics (fuel weight, which corresponds to CO_2 emissions, and NO_X emissions) and total operating cost. Simple models simulating the potential "greener technologies" are modeled in MATLAB [33] and then integrated with FLOPS for the performance calculations. The goal of the aircraft sizing problem is to develop an aircraft with 2940 nmi design range with a seat capacity of 162 seats in two classes. A brief description of the greener aircraft design optimization problem appears below. For more details about the aircraft design problem, we encourage the readers to see Ref. [38].

4.3.1 Description of the continuous variables

The problem includes ten continuous variables that define the wing and the engine parameters of the aircraft. The details of these continuous design variables and their design bounds appear in **Table 2**.

4.3.2 Simulating the discrete technologies

This aircraft design optimization study models three types of discrete technologies. **Table 3** lists the set of discrete technologies considered in this study. To model composite material selection choice on various aircraft components, the approach here uses a binary variable for each of the aircraft components that includes wing, fuselage, tail, and nacelle. A value of one represents composites being present while a value of zero represents no composite materials in that structure. The second discrete variable includes the eight possible combinations of the location and the number of engines. Lastly, eight combinations of laminar flow technologies are included for this problem, depending on whether it is natural laminar flow (NLF) or hybrid laminar flow control (HLFC) technology and the number of components on which it is applied (as listed in **Table 3**). References [38–40] describe the various discrete technologies used in this study in further detail.

Design variables	Lower bound	Upper bound
Aspect Ratio	8	12
Taper Ratio	0.3	0.5
Thickness to Chord Ratio	0.09	0.17
Wing Area [ft^2]	1,000	1,500
Wing Sweep at 25 percent [deg]	0	40
Thrust per engine [lbs]	20,000	30,000
By-Pass Ratio	5	10
Turbine Inlet Temperature [R]	3010	3510
Overall Pressure Ratio	35	55
Fan Pressure Ratio	1.6	1.7

Table 2.
Continuous variables for aircraft design problem.

Laminar Flow Technologies	Engine Position	Composite Material Choices			
		Wing	Fuselage	Nacelle	Tail
NLF-Wing	2 wing	Yes	Yes	Yes	Yes
HLFC-Wing	2 fuselage	No	No	No	No
HLFC-Wing + Nacelle	2 wing +1 fuselage				
HLFC-Wing + Tail	3 fuselage				
HLFC-Wing + Tail + Nacelle	4 wing				
NLF-Wing + HLFC-Tail	2 wing +2 fuselage				
NLF-Wing + HLFC-Nacelle	1 fuselage				
NLF-Wing + HLFC-Tail + HLFC-Nacelle	4 wing +1 fuselage				

Table 3.
Discrete technologies for aircraft design problem.

The problem also has four constraints that appear in **Table 4**. The constraints ensure that the design solution meets the desired field length criteria, has sufficient ground clearance, and sets a maximum limit on the amount of allowable fuel carrying space in the fuselage.

The aircraft design optimization problem considers two different pairs of competing objectives. The first pair involves simultaneous minimization of the aircraft fuel weight (index of CO_2 emissions) and the total operating cost of the aircraft, and the second pair involves minimizing the NO_X emissions and the total operating cost of the aircraft. The GA population has been limited to 48 individuals while setting the upper limit for the number of generations to 50 as before. The maximum

Take-off field length [ft]	≤ 8,000
Landing field length [ft]	≤ 7,500
Landing gear length [in]	≤ 150
Fuselage fuel capacity [lbs]	≤ 28,800

Table 4.
Problem constraints.

number of function evaluations for the SQP minimization (using MATLAB's fmincon) have been limited to the default value of 100 times the total number of continuous variables for this study. For certain combinations of discrete technology selection choices, the gradient-based approach may not find a feasible solution. In such cases, as mentioned in the methodology section, those designs are assigned high penalty for elimination in the subsequent generations.

4.3.3 Results for aircraft design problem

4.3.3.1 Objective pair - fuel weight vs. total operating cost.

Figure 9 shows the set of 24 non-dominated designs for the competing objective pair – aircraft fuel weight and total operating cost. The aircraft fuel weight, analogous to fuel burn, is directly related to the amount of CO_2 produced during the trip. The Pareto frontier consists of designs employing combinations of composite structures, eight different engine position(s), and eight different laminar flow technologies, modeled as a part of the greener technology approaches described in the previous section.

The design point ND1 (for Non-Dominated design number 1) in **Figure 9** corresponds to highest total operating cost (also lowest fuel weight) and makes use of NLF technology on the wing and HLFC technology on the nacelles and tail, along with two wing-mounted engines. This design also features composite wings, fuselage, and nacelles. The use of composite structures leads to a decrease in the fuel consumption (due to the reduction in aircraft empty weight) at the expense of increased total operating cost (due to increase in the manufacturing and maintenance costs associated with composite materials). The design with the lowest total operating cost (ND24) makes use of NLF technology on the wing and HLFC technology on the nacelles and tail, along with two wing-mounted engines as well. But, this design has no composite components and, hence, has the lowest total operating cost according to the models used in this study.

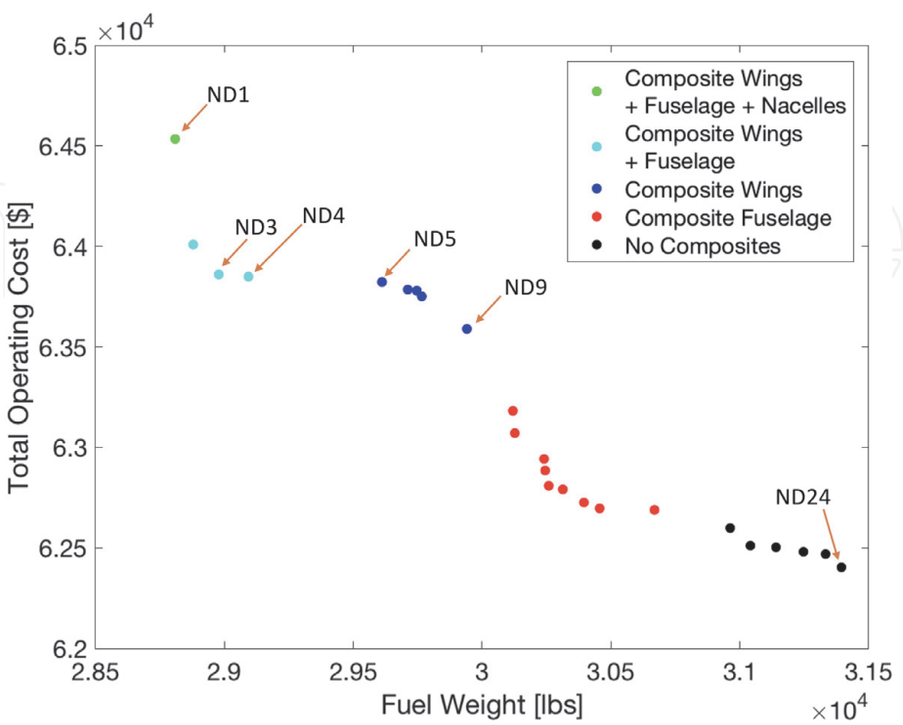


Figure 9.
The non-dominated set for objective pair – aircraft fuel weight (index for CO_2 production) and the total operating cost.

All the non-dominated designs employ NLF technology on the wings and HLFC technology on both the nacelle and tail, along with two wing-mounted engine configuration. The laminar flow technologies tend to reduce the skin friction drag of the aircraft, making the design more aerodynamically efficient, and reducing its fuel consumption for a given mission range. All the non-dominated designs employ these technologies in various forms (NLF or HLFC) to reduce fuel burn, depicting the importance of employing these technologies in near future "greener" aircraft design.

An interesting region in the Pareto frontier from an airline's standpoint would be near the points ND1 and ND3 (or ND2), where a substantial decrease in total operating cost is possible for a marginal increase in the aircraft fuel weight (index of CO_2 production per trip). Considering non-dominated designs ND1 and ND3, a nearly 1% reduction in total operating cost is possible to achieve for only a 0.6% increase in the total fuel weight needed for the mission, as one move from ND1 to ND3. Similar trends for the objective pair in consideration are also observed for designs ND9, ND10, and ND11.

4.3.3.2 Objective pair - NO_X emissions and total operating cost

The Pareto front corresponding to the NO_X emissions and the total operating cost objective pair appears in **Figure 10** and has 24 non-dominated designs. The non-dominated designs have different geometric design variable values that best match the different discrete greener aircraft technologies to arrive at the trade-off between the NO_X emissions and the total operating cost.

The design with minimum NO_X emissions and maximum total operating cost (ND1) employs a three-engine configuration with one fuselage-mounted and two wing-mounted engines, along with a composite wing. The laminar flow technologies on this design include NLF technology on wings and HLFC technology on the nacelles and tail. The maximum NO_X emitting design with minimum total operating cost (ND24) employs a two-engine configuration with wing-mounted engines, along with NLF technology on the wings and HLFC technology on the tail, and a composite nacelle. All the non-dominated designs, except the one with maximum NO_X emissions, employ NLF technology on the wings and HLFC technology on both the nacelle and tail. As we move from left to right along the Pareto frontier in

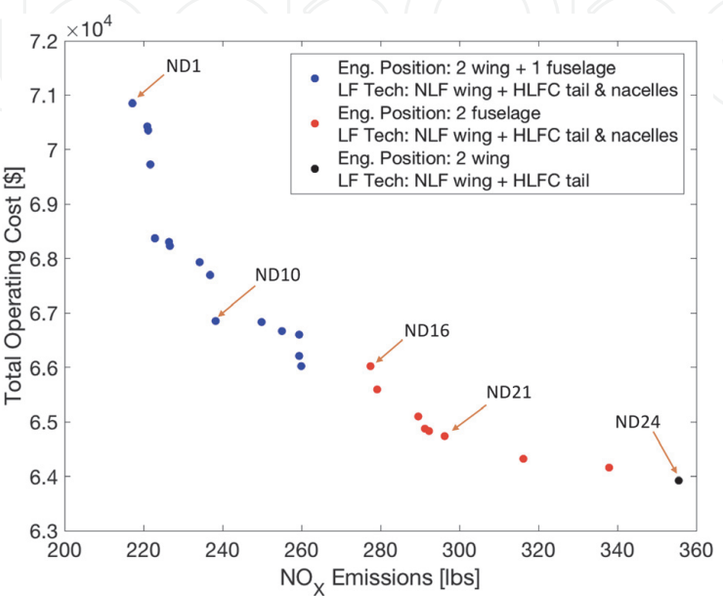


Figure 10.
The non-dominated set for objective pair – NOX emissions versus total operating cost.

Figure 10, the aircraft engines change from a three-engine configuration (two wing-mounted and one fuselage-mounted) to a two-engine (wing-mounted) configuration, thereby reducing the total operating cost.

An interesting region from the airline's point of view is the near the points ND2, ND3, ND4 and ND5, where a nearly vertical portion is visible in the top left portion of the Pareto frontier (refer to **Figure 10**). Moving from left to right in this region, a substantial decrease in total operating cost is possible for a marginal increase in the NO_X emissions of the aircraft. A plausible design from an airline's perspective–among the obtained non-dominated designs–would be the ND10 design. The reason for this observation is that a substantial increase in total operating cost will be incurred if further reduction in NO_X emissions are desired, while any effort to further reduce the total operating cost will lead to very high NO_X emissions, which is not desired from an environment standpoint.

Given there is some degree of randomness associated with the genetic operations in the GA, subsequent runs of the hybrid GA for the two objective pairs find a slightly different number of non-dominated designs points. However, the basic trait of the Pareto frontier, in terms of the discrete choices, did not alter; only the density of points in the Pareto frontier varied with different runs.

5. Conclusions

This chapter describes a hybrid multi-objective algorithm that makes use of an efficient gradient-based SQP algorithm for fitness evaluation inside a GA in a learning approach. The combination allows the GA to evolve a population of designs in the direction of the Pareto frontier while the SQP algorithm enforces constraints, eliminating the need for penalty multipliers or other special constraint handling methods and refines the values of the continuous design variables. The selective parent mixing and unique sets of goal point assignment to the individual lead to a distinct improvement in convergence and the quality of the Pareto frontier from a previous variation of this approach. When applied to various constrained MDNLP engineering design problems, the hybrid algorithm shows the ability to identify promising designs.

Although the ability of the hybrid approach to solve difficult constrained MDNLP problems is demonstrated in this chapter, the methodology relies heavily on the constraint enforcing ability and efficient searching of the continuous design space via the local gradient-based SQP algorithm that requires some estimates (either numerically or analytically) of the gradients of the objectives and the constraints with respect to the continuous design variables. A major advantage of a gradient-based approach besides being able to enforce the problem constraints (hence, the motivation to hybridize) is that the computational cost needed to compute the gradients is nearly independent of the number of design variables [41] when using adjoint-based methods to estimate the derivatives. This allows the gradient-based approach to efficiently solve problems with a very large number of design variables. However, if the objectives are encapsulated in a black-box function and are computationally very expensive to evaluate, then it may not be possible to directly implement a gradient-based search and may require a surrogate-based design optimization approach [40, 42, 43].

Nomenclature

α_i	Weight vector for the relative under/over-attainment of objective
$f_i(\mathrm{x})$	Value of the objective

f_i^G	Goal value for objective
$g_i(\mathrm{x})$	Nonlinear inequality constraints
γ	Attainment factor
$h_i(\mathrm{x})$	Nonlinear equality constraints
n	Population size
x_c	Continuous design variable
x_d	Discrete design variable
x^L	Design variable lower bound
x^U	Design variable upper bound

Abbreviations

EA	Evolutionary algorithm
GA	Generic algorithm
HLFC	Hybrid laminar flow control
MDNLP	Mixed-discrete nonlinear programming
ND	Non-dominated design
NLF	Natural laminar flow
NSGA	Non-dominated Sorted Genetic Algorithm
SPEA	Strength Pareto Evolutionary Algorithm
SQP	Sequential Quadratic Programming

IntechOpen

Author details

Satadru Roy†, William A. Crossley*† and Samarth Jain†
Purdue University, West Lafayette, USA

*Address all correspondence to: crossley@purdue.edu

† These authors contributed equally.

IntechOpen

© 2021 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. CC BY

References

[1] Cirillo R. The Economics of Vilfredo Pareto. Routledge; 1979. ISBN: 978-1-136-27816-7

[2] Coello Coello C A. A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques. Knowledge and Information Systems 1, 1999;269–308. DOI: 10.1007/BF03325101

[3] Deb K, Goel T. A Hybrid Multi-objective Evolutionary Approach to Engineering Shape Design. In: Zitzler E, Thiele L, Deb K, Coello Coello C A, and Corne D, editors. Evolutionary Multi-Criterion Optimization. Lecture Notes in Computer Science, vol 1993. Springer; 2001. p. 385-399. DOI: 10.1007/3-540-44719-9_27

[4] Deb K, Goel T. Controlled Elitist Non-dominated Sorting Genetic Algorithms for Better Convergence. In: Zitzler E, Thiele L, Deb K, Coello Coello C A, and Corne D, editors. Evolutionary Multi-Criterion Optimization. Lecture Notes in Computer Science, vol 1993. Springer;2001. p. 67-81. DOI: 10.1007/3-540-44719-9_5

[5] Holland J H. Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence. MIT Press;1992. 232 p. ISBN: 978-0-262-08213-6

[6] Sequential Quadratic Programming. In: Nocedal J, Wright S J, editors. Numerical Optimization. Springer Series in Operations Research and Financial Engineering. Springer;2006. p. 529-562. DOI: 10.1007/978-0-387-40065-5_18

[7] Fundamentals of Unconstrained Optimization. In: Nocedal J, Wright S J, editors. Numerical Optimization. Springer Series in Operations Research and Financial Engineering. Springer; 2006. p. 10-29. DOI: 10.1007/978-0-387-40065-5_2

[8] Schaffer J. Some Experiments in Machine Learning Using Vector Evaluated Genetic Algorithms [thesis]. 1985

[9] Fonseca C M, Fleming P J. Genetic Algorithms for Multiobjective Optimization: FormulationDiscussion and Generalization. In: Proceedings of the 5th International Conference on Genetic Algorithms; 1993. San Francisco: Morgan Kaufmann Publishers Inc.;1993. p. 416-423

[10] Zitzler E, Thiele L. An Evolutionary Algorithm for Multiobjective Optimization: The Strength Pareto Approach. 1998

[11] Srinivas N, Deb K. Muiltiobjective optimization using nondominated sorting in genetic algorithms. Evolutionary Computation. 1994; 221-248. DOI: 10.1162/evco.1994.2.3.221

[12] Coello C A. An updated survey of GA-based multiobjective optimization techniques. ACM Computing Surveys. 2000;109-143. DOI: 10.1145/358923.358929

[13] Konak A, Coit D W, Smith A E. Multi-objective optimization using genetic algorithms: A tutorial. Reliability Engineering & System Safety. Special Issue - Genetic Algorithms and Reliability. 2006;992-1007. DOI: 10.1016/j.ress.2005.11.018

[14] Goldberg D E. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley Publishing Company;1989. 372 p. ISBN: 978-0-201-15767-3

[15] Crossley, W A, Cook A M, Fanjoy D W, Venkayya V B. Using the Two-Branch Tournament Genetic Algorithm

for Multiobjective Design. AIAA Journal. 1999;37:261-267. DOI: 10.2514/2.699

[16] To T B, Korn U. MOBES: A Multiobjective Evolution Strategy for Constrained Optimization Problems. In: Proceedings of the third international conference on genetic algorithms;1997. p. 176-182

[17] Fonseca, C M, Fleming, P J. Multiobjective Optimization and Multiple Constraint Handling with Evolutionary Algorithms-Part I: A Unified Formulation. IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans. 1998;28:26-37. DOI: 10.1109/3468.650319

[18] Deb K, Agrawal S, Pratap A, Meyarivan, T. A Fast Elitist Non-dominated Sorting Genetic Algorithm for Multi-objective Optimization: NSGA-II. In: Schoenauer M, Deb K, Rudolph G, Yao X, Lutton E, Merelo J J, Schwefel H, editors. Parallel Problem Solving from Nature PPSN VI. Springer; 2000. p. 849-858. DOI: 10.1007/3-540-45356-3_83

[19] Whitley D, Gordon V S, Mathias Keith. Lamarckian evolution, the Baldwin effect and function optimization. In: Davidor Y, Schwefel H, Männer R, editors. Parallel Problem Solving from Nature — PPSN III. Springer;1994. p. 5-15. DOI: 10.1007/3-540-58484-6_245

[20] Houck C R, Joines J A, Kay M G. Utilizing Lamarckian Evolution and the Baldwin Effect in Hybrid Genetic Algorithms. 1996

[21] Xiao X, Dow E R, Eberhart R, Miled Z B, Oppelt R J. A hybrid self-organizing maps and particle swarm optimization approach. Concurrency and Computation: Practice and Experience. 2004;16:895-915. DOI: 10.1002/cpe.812

[22] Kumar A, Sharma D, Deb K. A hybrid multi-objective optimization procedure using PCX based NSGA-II and sequential quadratic programming 2007 IEEE Congress on Evolutionary Computation. 2007;ISSN: 1941-0026. p. 3011–3018. DOI:10.1109/CEC.2007.4424855

[23] Isaacs A, Ray T, Smith W. A Hybrid Evolutionary Algorithm With Simplex Local Search 2007 IEEE Congress on Evolutionary Computation. 2007; ISSN: 1941-0026. p. 1701-1708. DOI:10.1109/CEC.2007.4424678

[24] Satoru H, Tomoyuki H, Mitsunori M. Hybrid optimization using DIRECT, GA, and SQP for global exploration 2007 IEEE Congress on Evolutionary Computation. 2007; ISSN: 1941-002. p. 1709-1716 DOI:10.1109/CEC.2007.4424679

[25] Gil C, Márquez A, Baños R, Montoya M G, Gómez J. A hybrid method for solving multi-objective global optimization problems. Journal of Global Optimization. 2007;38:265-281. DOI: 10.1007/s10898-006-9105-1

[26] Majig M, Hedar A, Fukushima M. Hybrid evolutionary algorithm for solving general variational inequality problems. Journal of Global Optimization. 2007;38:637-651. DOI: 10.1007/s10898-006-9102-4

[27] Lehner S, Crossley W A. Hybrid Optimization for a Combinatorial Aircraft Design Problem In: 9th AIAA Aviation Technology, Integration, and Operations Conference (ATIO); 2009. DOI: 10.2514/6.2009-7116

[28] Zhong X, Fan W, Lin J, Zhao Z. Hybrid Non-dominated Sorting Differential Evolutionary Algorithm with Nelder-Mead. Second WRI Global Congress on Intelligent Systems. 2010;1: 306-311. DOI: 10.1109/GCIS.2010.198

[29] Wang X, Tang L. A PSO-Based Hybrid Multi-Objective Algorithm for Multi-Objective Optimization Problems.

In: Tan Y, Shi Y, Chai Y, Wang G, editors. Advances in Swarm Intelligence. ICSI 2011. Lecture Notes in Computer Science, vol 6729, Berlin, Heidelberg: Springer; 2011. DOI: 10.1007/978-3-642-21524-7_4

[30] Li X, Du G. BSTBGA: A hybrid genetic algorithm for constrained multi-objective optimization problems. Computers & Operations Research. 2013;40:282-302. DOI: 10.1016/j.cor.2012.07.014

[31] Žilinskas A, Žilinskas J. A hybrid global optimization algorithm for non-linear least squares regression. Journal of Global Optimization. 2013;56:265-277. DOI: 10.1007/s10898-011-9840-9

[32] Lohpetch D, Jaengchuea S. A Hybrid Multi-objective Genetic Algorithm with a New Local Search Approach for Solving the Post Enrolment Based Course Timetabling Problem Advances in Intelligent Systems and Computing. Springer International Publishing. 2016; 195-206 DOI: 10.1007/978-3-319-40415-8_19

[33] The Mathworks Inc., MATLAB version 2017a

[34] MathWorks Solve multiobjective goal attainment problems - MATLAB fgoalattain https://www.mathworks.com/help/optim/ug/fgoalattain.html [Accessed 4 February 2020]

[35] Roy S. Multi-objective Optimization using a Hybrid Approach for Constrained Mixed Discrete Non-linear Programming Problems - Applied to the Search for Greener Aircraft [thesis]. Purdue University; 2012.

[36] Jain S. A Multi-Fidelity Approach to Address Multi-Objective Constrained Mixed-Discrete Nonlinear Programming Problems With Application to Greener Aircraft Design [thesis]. Purdue University; 2018.

[37] McCullers L A, FLOPS, Software Package, Ver. 8.12 NASA Langley Research Center. Hampton, VA 2010

[38] Lehner S, Crossley W. A. Combinatorial Optimization to Include Greener Technologies in a Short-to-Medium Range Commercial Aircraft In: The 26th Congress of ICAS and 8th AIAA ATIO DOI: 10.2514/6.2008-8963

[39] Roy S, Crossley W A. Hybrid Multi-Objective Combinatorial Optimization Technique with Improved Compatibility between GA and Gradient-Based Local Search. In: Proceedings of 12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference; September 2012. AIAA; 2012

[40] Jain S, Crossley W A, Roy S. A Multi-Fidelity Approach to Address Multi-Objective Mixed-Discrete Nonlinear Programming Problems. In: 2018 Multidisciplinary Analysis and Optimization Conference; June 2018. AIAA; 2018

[41] Gray J S, Hwang J T, Martins J R R A, Moore K T, Naylor B A. OpenMDAO: An open-source framework for multidisciplinary design, analysis, and optimization. Structural and Multidisciplinary Optimization. 2019; 59:1075-1104. DOI: 10.1007/s00158-019-02211-z

[42] Jones D R, Schonlau M, Welch W J. Efficient Global Optimization of Expensive Black-Box Functions. Journal of Global Optimization. 1998;13:455-492. DOI: 10.1023/A:1008306431147

[43] Roy S, Crossley W A, Moore K T, Gray J S, Martins J R R A. Monolithic Approach for Next-Generation Aircraft Design Considering Airline Operations and Economics. Journal of Aircraft. 2019;56:1565-1576. DOI: 10.2514/1.C035312